A complete solution for duplication detection over uncertain data

Peng Pan*, Xiaojun Cai

School of Computer Science and Technology, Shandong University, Jinan, P R China

Received 1 March 2014, www.cmnt.lv

Abstract

As the problem of uncertainty for duplication is increasingly prominent with the sharp growth of amount and scale for data sources, we need to pay more attention on it. However, the research on uncertainty about duplicated data is still on its start. In this paper, we propose a complete method for duplication detection with probability, which is efficient and suitable for large-scale dataset. Considering the large-scale background, firstly, we adopt the rapid cluster algorithm based on canopies to get blocks. Secondly, in order to generate the record sets, which represent entity, we provide one fuzzy cluster method over each block by assigning two thresholds. By doing these, we balance the complexity and accuracy. Finally, we assign the probability for each record in one block. The experiments show advantages over other present algorithms for performances.

Keywords: duplication detection, data uncertainty, canopy, data probability

1 Introduction

Duplication detection and data fusion are challenges in data integration. Duplication mainly arises from these situations: one entity has various expressions in different data sources owing to the heterogeneous schemas and semantics; two records might describe the different aspects for the same entity in one integrated data source, which has solved the problem of isomerism for schema and semantics. These issues will result in a large amount of fuzzy subordinations, which imply the uncertainty.

It is a tough task to handle data accurately for current methods of duplication detection and data fusion because one complete domain knowledge cannot be acquired easily, and the contents of duplicated records are usually inconsistent, incomplete and inaccurate. Therefore, manual interventions are necessary to improve the accuracy. However, it is not practicable for artificial means in large-scale data environments such as deep web. While automatic method is adopted to improve the efficiency, it usually choose the most possible information with the loss of some useful parts. These methods are not capable of guaranteeing the quality of duplication detection and data fusion.

As the problem of uncertainty for duplication is increasingly prominent with the sharp growth of amount and scale for data sources, we need to pay more attention on it. However, the research on uncertainty about duplicated data is still on its start. [1] builds a model for probabilistic database of duplicated data, and provides one query method based on the model. [2] provides a method for generating the probabilistic database over dataset. [3] proposes a algorithm for probabilistic duplication detection based on graph theory, but the complexity is so high that it is not suitable to be used in large scale dataset.

In this paper, we propose a method for duplication detection with probability, which is efficient and fit in large-scale dataset. Firstly, we design one algorithm for rapid blocking based on canopies to get a lot of block called canopy. Secondly, in order to generate the records set which represents an entity, we provide one fuzzy cluster method over each block by assigning two thresholds. Finally, we assign the probability for each record in one block.

The contributions for this paper are:

- Considering the large-scale background, we carry out the rapid cluster algorithm based on canopies, and then adopt fuzzy cluster method with two thresholds. By doing this, we balance the complexity and accuracy.
- We also provide one method to assign the probability for each record, whose experiment shows high efficiency.

2 Related works

For the uncertainty of duplication, it means that which records from different data sources are put together is uncertain, and what is the representative of one entity in one record set representing the entity is uncertain. For these uncertainty, [4] defines one "integration" operation to handle conflicting records, for example, the ages for one person in two relation are 23 and 24, respectively. The output is [23, 24], and each value has probability. [5] defines a data model to fuse the data tree expressed in XML, and assigns probability for each representative with one method called "frenquenlistic". The similar method

* ***Corresponding author*** e-mail: ppan@sdu.edu.cn

appears in [6] and [7], where XML is assigned with probability, but the amount of representatives is reduced by outside domain knowledge. [8] provides a language to express the integrated results with uncertainty. [9] proposes a methods for generating probabilistic database over duplicated data. It also provides the algorithm to obtain the representative in the records set representing one entity, and assign one probability for each record. However, it does not explain how to obtain the original records. [10] generates one probabilistic database for duplicated data by hierarchical clustering with different parameters , and provides one effective query method over it. [3] provides the probabilistic duplication detection method based graph theory, but it has so high complexity that cannot be used in large scale data. [2] provides several algorithm for cluster to generate duplicated records set with uncertainty, and compares the methods for probability assignment.

3 Constructing possible sets over duplicated data clusters

3.1 BLOCK FOR MASSIVE DATA

For large scale datasets, it is inefficient for applying traditional cluster methods to construct huge matrix. Especially in real-time environments such as deep web query, the problem on how to improve the efficiency for cluster algorithms has been urgent. Recently, many methods such as Sorted Neighbourhood [11], Bigram Indexing etc. are proposed to solve the problem of large-scale cluster. In this paper, we adopt the idea of canopies [12] to improve the efficiency for duplication detection.

The process of blocking cluster data by canopies has two steps: firstly, one rough and low cost methods is applied to divide the source data into some overlapped subsets called canopies, whose certain data is the centre in the range. Secondly, canopies are one clustering algorithm with higher cost and calculations that are more accurate.

The main idea in this paper for canopies firstly find out all the data around one centre to create one canopy with minimal cost, then find the domain for next centre to create another canopy, this process iterates till all the data are included in canopies. Since canopies are overlapped, one data might exist in more than one canopy. Therefore, in order to guarantee all the data exist in canopies, we use two threshold: τ_1, τ_2 and $\tau_1 \geq \tau_2$. Algorithm 1 is the detailed description.

Algorithm 1: The rapid block methods based on canopies

INPUT: data source D OUTPUT: canopies

1 $CenterSet \leftarrow$ D
2 $i = 0$
3 while $CenterSet \neq \phi$ do
4 $d \leftarrow \arg\min_{d \in CenterSet}(approxDist(d, d'))$
5 $\text{canopy}_i(d) \leftarrow \{ d' \mid d' \in D \wedge approxDist(d, d') \leq \tau_1 \} \cup \{d\}$
6 $CenterSet \leftarrow CenterSe\text{-}\{d' \mid d' \in \wedge approxDist(d,d') \leq \tau_2 \} \cup \{d\}$
7 $i = i + 1$
8 enddo

CenterSet is a candidate data centre point set, whose initial value is the whole data set. When *CenterSet* is null, the algorithm will end; the second to seventh lines describe the process for generating one canopy, and the fourth line select one data point *d* as the centre from two data, which form the shorted distance, and *approxDist* is the algorithm for rapid calculating the distances. The fifth line puts those data whose distances to the centre d are less than τ_1 into the canopy whose centre is *d*; the sixth line remove original centre and the data whose distances to the centre *d* are less than τ_2 from the centre data point set. Hence, these removed points are regarded as the centre points set of the canopy in this iteration, and this can guarantee each data point exists in only one centre points set of canopy. When *CenterSet* is null, it implies that all the data have been put into the canopies.

3.2 THE CLUSTER PROCESS IN BLOCKS

In each block formed by canopies, we adopt more fine clustering algorithm to generate cluster divisions. As a result, each division stands for one entity, the data indicating the same entity will be in one same cluster division. This paper divides each cluster division into two parts by two thresholds: core and edge. Among them, the core part is constituted by data with high similarity value, which is above the ceiling threshold θ_1 and the edge part consists of that with lower similarity value, which is between the bottom threshold θ_2 and ceiling threshold θ_1. Each data appears in one core part for only once, but can appear in more than one edge parts. Algorithm 2 is the detailed description.

Algorithm 2: The Clustering in canopy

INPUT: canopy S,
The similarity pair G in S ,
Thresholds: τ_1 , τ_2
OUTPUT: final cluster divisions set C_f

1 $M \leftarrow G$
2 $C_s \leftarrow \emptyset,\ C_f \leftarrow \emptyset$
3 $CC \leftarrow \emptyset,\ CM \leftarrow \emptyset$
4 i=0
5 while $\max_{sim(w,v) \in M}(sim(w,v)) \geq \tau_2$ do
6 $u \leftarrow \arg\max_{u \in \{w \mid sim(w,v) \in M\}}(sim(u,v))$
7 $CC_i \leftarrow \{w \mid sim(u,w) \geq \tau_1 \wedge w \notin C_s\} \cup \{u\}$
8 $CM_i \leftarrow \{w \mid sim(u,w) < \tau_1 \wedge sim(u,w) \geq \tau_2 \wedge w \in S\}$
9 $M \leftarrow M - \{sim(u,v) \mid v \in CC_i \wedge sim(u,v) \in M\}$
10 $C_s \leftarrow C_s \cup CC_i$
11 $CC \leftarrow CC \cup \{CC_i\}$
12 $CM \leftarrow CM \cup \{CM_i\}$
13 $C_f \leftarrow C_f \cup \{\{CC_i\}, \{CM_i\}\}$
14 i = i + 1
15 enddo

M is the table for similarity, which records the similarity value for all the data in one canopy; C_s is the core data nodes at present; CC is the core cluster set, whose

format is $\{\{CC_1\},\{CC_2\},\ldots\{CC_k\}\}$. *CM* is the edge cluster set, whose format is $\{\{CM_1\},\{CM_2\},\ldots\{CM_k\}\}$, C_f is the final cluster divisions, whose format is $\{\{\{CC_1\}, \{CM_1\}\}, \{\{CC_2\}, \{CM_2\}\} \ldots\{\{CC_k\},\{CM_k\} \}\}$.

The fifth to fourteenth lines describe the cluster process. The sixth line selects the pair(u,v) with maximum similarity from *M*, and set *u* as base. The seventh line puts all the data nodes whose similarities value with *u* are more than τ_1 into one core cluster division CC_i. The eighth line puts all the data nodes whose similarity with *u* is more than τ_2 and less than τ_1 into one edge cluster division CM_i. Since *v* and *u* have been put into core cluster division, the ninth line removes the similarity about *v* and *u*. The tenth and thirteenth lines update the sets C_s,*CC*,*CM*,C_f.

3.3 THE CALCULATION FOR PROBABILITY OF ELEMENTS IN CLUSTER DIVISIONS

The probability for one element in cluster division stands for the chance for which the element exists in one clean instance potentially. The method for calculating the probability has three steps:

1. Acquires the representative element *rep*.
2. Computes the sum d of distance between *rep* and each element in division.
3. Represents probability with $\frac{d}{\sum d}$.

Algorithm 3 is the description in detail.

Algorithms 3: The calculation for probability for elements in cluster divisions

INPUT: a set of records *R*,
Cluster *C* over *R*,
a similarity function *sim*()
OUTPUT: a set of probability *P*

1 for each $C_i \in C$ do
2 $C^* \leftarrow \emptyset$
3 for each $r \in C_i$ do
4 $rep = \arg\max_{r \in C_i}(\sum_{s \in C_i} sim(r,s))$
5 for each $t \in C_i$ do
6 $p(t) = \frac{sim(t,rep)}{\sum_{r \in C_i} sim(r,rep)}$

The fourth line accomplishes step 1) and the fourth line finishes step 2) and 3). We adopt Softtf-idf [13] method with *q*-grams as the similarity function

4 Experiments

We conduct two experiments to evaluate the performances of the methods proposed in this paper:

1) We compare the performances between the algorithm with canopy and one without canopy over the same dataset

2) We evaluate the performances for various algorithms for probability assignments.

We have collected 9978 book records from 50 online book shop, and 1879 records is regard as the final dataset for experiments after manual tagging.

4.1 COMPARE THE PERFORMANCES BETWEEN THE ALOGIRTHM WITH CANOPY AND ONE WITHOUT CANOPY OVER LARGE SCALE DATASET

4.1.1 Experiment Design

We handle the dataset by the cluster algorithm with canopies and without canopies, where the latter is the method proposed in 3.2. We set the threshold for the two algorithms to 0.75, and compare the precision ration, recall ratio and executing time over the results.

4.1.2 Evaluation Criteria

We given the calculating methods for precision ration and recall ratio.

Suppose we have a exact cluster $G=\{g_1,\ldots g_k\}$ over relation *R*, let $C=\{c_1,\ldots c_k^{'}\}$ is the *k-th* output cluster by clustering algorithm. We define a mapping function *f* from *G* to *C*, which maps each exact cluster g_i into one output cluster c_j, i.e. $c_j = f(g_i)$. Therefore, the precision ration and recall ratio for one cluster g_i are defined as following:
The precision ratio of a single cluster g_i:

$$Prec_i = \frac{|f(g_i) \cap g_i|}{|f(g_i)|}.$$

The recall ratio of a single cluster g_i:

$$Recl_i = \frac{|f(g_i) \cap g_i|}{|g_i|}.$$

As far a clustering algorithm as be concerned, its precision ration and recall ratio can be defined as the weighted average, which is defined as following:

The precision ratio of all clusters:

$$Prec = \frac{|G \cap C|}{|G|} = \sum_{i=1}^{k} \frac{|g_i|}{|R|} Prec_i$$

The recall ratio of all clusters:

$$Recl = \frac{|G \cap C|}{|C|} = \sum_{i=1}^{k} \frac{|g_i|}{|R|} Recl_i.$$

In addition, we define the harmonic methods F_1, which is formulated as $F_1 = \frac{2 \times Prec \times Recl}{Prec + Recl}$.

4.1.3 Results

Table 1 shows that the method with canopies has lower precision ratio and recall ratio than the algorithm without canopies, but has much lower execution time than the

latter. Taken together, the method with canopies is more preferable for large scale dataset.

TABLE 1 The performances compare

	Precision	Recall	F1	Execution Times(ms)
Canopies	0.709	0.964	0.817	3421
without Canopies	0.735	0.987	0.843	124852

4.2 PREFORMANCES FOR VARIOUS ALGORITHMS FOR PROBABILITY ASSIGNMENTS

4.2.1 Experiment Design

We compare the similarity measurements for Wjaccard, SoftTfIdf, Cosine w/tfidf by adopting the probability assignment proposed in 3.3. We randomly select ten clusters, and conduct these measurements respectively.

4.2.2 Experiment Criteria

We use the order parameter promise ration (*OPR*) to evaluate the influence of probability assignment over probability order. The calculation is following:

Suppose the right order for probability value in one records set is $L_{correct}$, we denote the probability value of record r as $p(r)$; the order for probability according to certain function is L_{output}, where the probability value of record r as $po(r)$. By computing the amount of pair (r_i,r_j) for which r_i and r_j appear in $L_{correct}$ and L_{output} together, we use the order parameter promise ratio (*OPR*) to evaluate the extent to which the probability assignment algorithm by one function retains the original order. The computing equation is $OPR=\frac{|(ri,rj)\,|\,r_i,r_j \in L_{output}, i \le j, p(r_i) \le p(r_j)|}{C_k^2}$,

where C_k^2 is all the pair of L_{output}.

4.2.3 Results

Figures 1 and 2 show that the *OPR* of SoftTfIdf method is close to Wjaccard, and higher than Cosine w/tfidf, and the execution time is far lower than the other two methods.

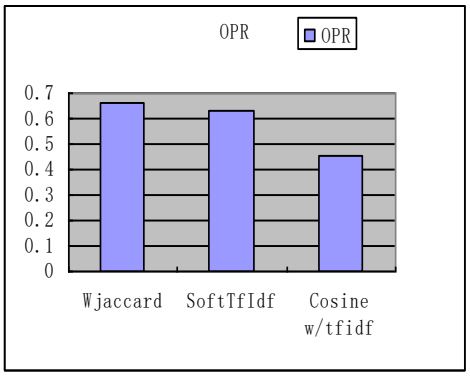

FIGURE 1 OPR for various algorithms for probability assignments

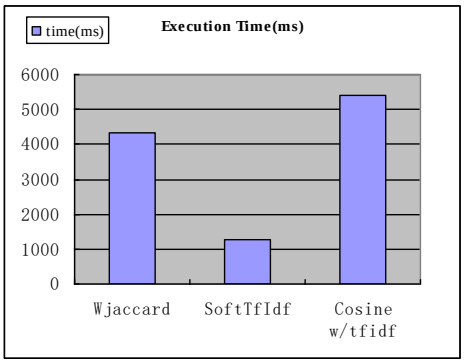

FIGURE 2 Execution Time for various algorithms for probability assignments

5 Conclusion

In this paper, we propose a complete method for duplication detection with probability considering the large scale background, firstly, we carry out the rapid cluster algorithm based on canopies. Secondly, in order to generate the records set which represents an entity, we provide one fuzzy cluster method over each block by assigning two thresholds. By doing these, we balance the complexity and accuracy. Finally, we assign the probability for each record in one block. The experiments show advantage over other present algorithms for performances.

References

[1] Beskales G, Soliman M A, Ilyas I F, Ben-David S 2009 Modeling and Querying Possible Repairs in Duplicate Detection *Proceedings of the VLDB Endowment* **2**(1) 598-609
[2] Hassanzadeh O, M R J 2009 Creating probabilistic databases from duplicated data *The VLDB Journal* **18**(5) 1141-66
[3] Panse F, van Keulen M, Ritter N 2010 Indeterministic Handing of Uncertain Decisions in Duplicate Detection *Technical report* University of Twente (Netherlands) TR-CTIT-10-21
[4] Tseng F S C, Chen A L P, Yang W 1993 Answering heterogeneous database queries with degrees of uncertainty *Distributed and Parallel Databases* **1**(3) 281-302
[5] van Keulen M, de Keijzer A, Alink W 2005 A probabilistic XML approach to data integration *Proceedings of the 21st International Conference on Data* Engineering 2005 ICDE 2005 459-70
[6] Hunter A, Liu W 2006 Fusion rules for merging uncertain information *Information Fusion* **7**(1) 97-134
[7] Hunter A, Liu W 2006 Merging uncertain information with semantic heterogeneity in XML *Knowledge and Information Systems* **9**(2) 230-58
[8] Cali A, Lukasiewicz T 2006 An approach to probabilistic data integration for the semantic Web *Uncertainty Reasoning for the Semantic Web I Lecture Notes in Computer Science* **5327** Springer-Verlag Berlin Heidelberg 52-65
[9] Andritsos P, Fuxman A, Miller R J 2006 Clean Answers over Dirty Databases:A probabilistic Approach *Proceedings of the 22nd International Conference on Data Engineering* 2006 30
[10] Gupta R and Sarawagi S 2006 Creating probabilistic databases from information extraction models *Proceedings of the 32nd international conference on Very large data bases* 965-76
[11] Baxter R , Christen P, Churches T 2003 A comparison of fast blocking methods for record linkage *ACM SIGKDD Workshop on Data Cleaning, Record Linkage, and Object Identification* 25-7

[12] McCallum A, Nigam K and Ungar L H 2000 Efficient clustering of high-dimensional data sets with application to reference matching *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* 169-178

[13] Cohen W, Ravikumar P, Fienberg S 2003 A comparison of string distance metrics for name-matching tasks *Proceedings of International Joint Conference on Artificial Intelligence* 73–78

Authors

Peng Pan, born in January, 1974, Anqiu, Shandong, China

Current position, grades: Lecture, Doctor of Computer Science.
University studies: Ph.D degree from Shandong University in 2010.
Scientific interest: deep web, data uncertainty, electronic commerce.
Publications: about 20.

Xiaojun Cai, born in September, 1976, Yizheng, Jiangsu, China

Current position, grades: lecturer, Doctor Candidate.
University studies: Shandong University.
Scientific interest: embedded technology, electronic commerce, data management.
Publications: about 10.